LA-UR -91-324

LA-UR--91-324

DE91 007478

TITLE Using CNLS-Net to Predict the Mackey-Glass Chaotic Time Series

AUTHOR(S) W. C. Mead, R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, L. A. Lee, and M. K. O'Rourke

SUBMITTED TO IJCNN-91-Seattle Conference
Seattle, WA

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Using CNLS-Net to Predict the Mackey-Glass Chaotic Time Series

W. C. Mead[a,b], R. D. Jones[a,b], Y. C. Lee,[b,c], C. W. Barnes[a,b]
G. W. Flake[a,b,d], L. A. Lee[a,b], and M. K. O'Rourke[a,b]

[a]*Applied Theoretical Physics Division*
[b]*Center for Nonlinear Studies*
University of California
Los Alamos National Laboratory, Los Alamos, NM 87545

[c]*Department of Physics and Astronomy*
[d]*Department of Computer Science*
University of Maryland, College Park, MD 20740

January 21, 1991

ABSTRACT

We use the Connectionist Normalized Local Spline (CNLS) network to learn the dynamics of the Mackey-Glass time-delay differential equation, for the case $\tau = 30$. We show the optimum network operating mode and determine the accuracy and robustness of predictions. We obtain predictions of varying accuracy using some 2 – 120 minutes of execution time on a **Sun SPARC-1** workstation. CNLS-net is capable of very good performance in predicting the Mackey-Glass time series.

Introduction

The Connectionist Normalized Local Spline (CNLS) network[1] combines a number of appealing features to yield a capable, versatile adaptive-computing network. The net features normalized radial basis functions, a linear gradient term, and simple, rapid solution of the training algorithm, plus a variety of optional capabilities including Kalman Noise Filtering.[2] The CNLS network has been successfully applied to a number of fitting, prediction, and control test examples, including a preliminary test of the net's ability to predict the Mackey-Glass equation.[2]

The Mackey-Glass (M-G) equation[3] is a time-delay ordinary differential equation that displays well-understood chaotic behavior with dimensionality dependent upon the chosen value of the delay parameter.[4] The time series generated by the M-G equation has been used as a test bed for a number of new adaptive computing techniques: a local linear (or quadratic) approximation method,[5] a back-propagation neural network,[6] and at least two radial basis function approaches.[7,8] In this work, we have performed extensive studies on the use of the CNLS-net to model and predict the Mackey-Glass time series.

The CNLS-Network Architecture

The Connectionist Nonlinear Local Spline Network (CNLS-net) was developed as an extension of previous adaptive network experience.[6,7,9,10] A natural evolution is to modify radial basis function (RBF) nets in a manner that improves interpolation and reduces the amount of training necessary for accurate learning.[1,11] CNLS-net architecture has a single hidden layer and starts from the identity

$$g(\vec{r}) = \frac{\sum_{j=1}^{N} g(\vec{r})\rho_j(\vec{r})}{\sum_j \rho_j(\vec{r})} \tag{1}$$

Here, as in the RBF network, $\rho_j(\vec{r})$ is a localized function of $\vec{r}$ about some $\vec{r}_j$. Hence, $g(\vec{r})$ on the right of Eq. 1 can be approximated by its Taylor expansion about $\vec{r}_j$. We have then,

$$\phi(\vec{r}) = \sum_{j=1}^{N} [f_j + (\vec{r} - \vec{r}_j) \cdot \vec{d}_j] \frac{\rho_j(\vec{r})}{\sum_j \rho_j(\vec{r})} \tag{2}$$

for an approximation to $g(\vec{x})$. This net differs from the RBF net in two ways: (1) the use of basis-function normalization and (2) the addition of a linear term, $(\vec{x} - \vec{x}_j) \cdot \vec{d}_j$. The use of a normalization term was suggested but not pursued by Moody and Darken.ry7 The addition of these two terms is responsible for the reduction in the amount of training data needed to obtain reasonable approximations. As in the case with radial basis functions, the training of f_j and $\vec{d}_j$ is linear and hence very fast.

The Mackey-Glass Equation

The Mackey-Glass (M-G) equation was first advanced as a model of white blood cell production.ry3 It is a time-delay differential equation, namely

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{[1 + x^c(t-\tau)]} - bx(t), \tag{3}$$

where the constants are often (and in this work) taken to be $a = 0.2, b = 0.1$, and $c = 10$. The behavior of the M-G equation as a function of the delay parameter τ has been studied extensively and is reported by J. D. Farmer in Ref. ry4. At $\tau = 30$, the value used for all of the studies reported here, the M-G equation's attractor has an information dimension of 3.6.ry4 Figure 1 shows a plot of the M-G equation with $\tau = 30$, slightly renormalized to limit its range approximately to the interval $(0, 1)$. The standard deviation of the function so normalized is 0.24. In the results presented below, we use as a performance indicator, the "Error Index," defined as the root mean squared fitting or prediction error (RMSE) divided by the standard deviation. With this definition of the Error Index, a constant fit through the mean value of the function leads to a value of 1.0.

Initial Choices for Embedding, Data sets, and Architecture

The starting place for this work was determined in large measure by the desire to compare CNLS-net's performance with that of the back propagation net used by Lapedes.[4] Thus, we initially chose the embedding used in previous worksry5-7: the training and test patterns were composed of 6 inputs, spaced at time intervals of 6 time units each, plus a test output, the point 6 time units after the last entry of the input sequence. Later, we found that embedding is a very sensitive matter, and performed more detailed studies of the issue.

The training and test files consisted of 1000 – 5000 points at fixed time spacing (usually, $\Delta t_{data} = 1$). The training and test files were non-overlapping time sequences, with the test file usually continuing the series begun in the training file. Except where noted, we used 500 training patterns, and we always used 500 test patterns, as did Lapedes.ry6 The training patterns were selected at random and the test patterns sequentially. Also, the selected training patterns were held fixed for the entire training period, but were usually "tumbled," *i.e.*, presented in random, varying sequences for successive training epochs.

The CNLS-net architecture chosen initially used 6 input nodes, 28 hidden nodes (having 7 adjustable weights each), and one output node. This yields about 200 weights, fewer than Lapedes' reported back-propagation net calculation: he used two hidden layers of 14 nodes each, giving about 540 weights, total.ry6 Our initial architecture yielded a network that could be trained in about 2 – 6 minutes and tested in about 1 minute (at 21 iterations into the future), which made multiparameter optimization feasible.

Optimizing CNLS-Net's Parameters

We performed an extensive optimization of the adjustable parameters of CNLS-net: the learning rate, the width of the basis functions, and the discreet parameters governing embedding and network size. We also devised two specific tests of versatility and robustness.

CNLS-net has two continuously-adjustable parameters. The learning rate shows a broad optimum, and learning behavior showed some regions of instability. We found a broad range of acceptable performance. Generally, higher learning rates led to faster training, with increased sus- ;)tability to instability. The width of the basis functions showed a broad optimum, as well. The optimum width appears to be related to the characteristic structures of the function being fit, under the chosen embedding.

The embedding structure is determined by the number of inputs to the network and the time-separation between each input. Results of the two-dimensional embedding study show that there is a lot of "structure." Small changes in the embedding integers lead to prediction errors that differ by a factor of $2\times$ or more. This is an important issue that needs to be understood better. We can summarize the embedding results by noting that most of the "successful" embeddings have Δt_{samp} in the range of $30-45$. We suspect this is related to the choice of $\tau = 30$ as the Mackey-Glass delay parameter. The parameter τ sets a "coherence timescale," and embeddings for which Δt_{samp} differs greatly from this time interval are either supplying the network with too little or too much information.

An example of the M-G time series and CNLS-net's fit is shown in Fig. 1. The plot of Fig. 2 shows the net's training and prediction accuracy as a function of training epoch. The net's prediction accuracy shows a broad optimum, but the net can be either over- of under-trained. We also found that trainability and prediction accuracy were influenced by the random initial choice of the basis function centers. Some of this sensitivity may be exacerbated by marginal stability of the learning algorithm, while some of the variation is due to the small numbers of basis functions used and the statistical effects of redistributing them.

Best Results

Our best results are compared with Lapedes' successful prediction in Fig. 3. The plot shows three sets of CNLS-net predictions, made with hidden layer sizes of $28, 56$, and 112 nodes.

The Mackey-Glass, $\tau = 30$ calculation of Lapedes and Farber[v6] was trained for about 60 minutes on a **Cray XMP** with vectorized coding. Our calculations were performed using **CNLSTOOL**, written in the C language, and executed on a **Sun SPARC-1** workstation. We estimate, without detailed, specific code measurements, that a speed conversion factor of about $40\times$ is probably about right between the two computers. Thus, our longest run, with predictive accuracy exceeding that of Lapedes' back propagation network, and with its training time of 2 hours, represents about a factor of $20\times$ improvement in computing resource requirement. Our faster runs, which of course are considerably less accurate in longer-time predictions, showed an additional factor of $20-60\times$ speedup, thus requiring fairly modest computing resources.

Versatility and Robustness

General concepts of versatility and robustness for numerical algorithms exist. For the purposes of this section, we qualitatively define "versatility" as the extent of the domain over which the network achieves "near-optimum" accuracy. We define "robustness" as the level of performance fluctuations, *e.g.*, fluctuations in prediction accuracy. We devised two ways of testing the versatility and robustness of the network: (1) changing the sampling time interval in the training and test files and (2) changing the time delay parameter in the Mackey-Glass equation. These tests can be made more or less sensitive by adjusting the continuous parameter excursions to match the versatility of the network being tested.

Here, we discuss the first test, varying the time spacing of data points in the training and test files. This test provides a continuous handle on the matching of effective feature size or "wavelength" of the function to be fit and the network basis functions. Figure 4 shows the results for three different network sizes. For a fixed set of network parameters, but retraining the net for each Δt_{data}, we find that variations of order 10% in point spacing significantly affect the ability of the network to obtain predictive fits. Increasing the network size gradually improves the net's versatility and robustness, both on this test and in our other, related calculations.

Conclusions

CNLS-net has proven able to accurately predict the behavior of the Mackey-Glass equation. We have obtained predictions that match the accuracy of previous work,[5,6] while requiring about 20× lower computational effort.[6] Data requirements are comparable with those of a back propagation network[6] and much less than those of unnormalized radial basis function nets.[7] Overall, CNLS-net's use of normalized, localized basis functions with linearized correction terms appears to be a successful approach. Qualitatively, in this low-dimensional space, CNLS-net behaves in ways that are roughly intermediate between typical back propagation (BP) networks and radial basis function (RBF) nets. CNLS-net learns much faster than a BP net, but becomes more readily confused in cases of high dimensionality or with excess data to analyze. CNLS-net requires less training data than RBF nets. Accuracy generally improves with larger networks, larger training sets, and greater training times. Versatility and robustness of the net improve somewhat with network size.

ACKNOWLEDGEMENTS

The authors appreciate the support of the Center for Nonlinear Studies, the Inertial Fusion and Plasma Theory Group (X-1), the Applied Theoretical Physics Division (X), and Los Alamos National Laboratory. This work was supported in part by the U.S.D.O.E.

REFERENCES

[1] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, *et al.*, "Function Approximation and Time Series Prediction with Neural Networks," Los Alamos National Laboratory Rpt. No. LA-UR-90-21 (1990).

[2] R. D. Jones, Y. C. Lee, S. Qian, C. W. Barnes, *et al.*, "Nonlinear Adaptive Networks: a Little Theory, a Few Applications," in preparation (1990).

[3] M. C. Mackey and L. Glass, Science **197**, 287 (1977).

[4] J. D. Farmer, Physica **4D**, 366 (1982).

[5] J. D. Farmer and J. J. Sidorowich, *Phys. Rev. Lett.* **59**, 845 (1987).

[6] A. Lapedes and R. Farber, Los Alamos National Laboratory Rpt. No. LA-UR-87-2662 (1987).

[7] J. Moody and C. J. Darken, Neural Comp. **1**, 281 (1989).

[8] M. Casdagli, Physica D **35**, 335 (1989).

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in **Parallel Distributed Processing, Vol. 1**, eds. D. E. Rumelhart and J. L. McClelland (MIT, Cambridge, 1986), pp. 318-362.

[10] P. D. Wasserman, **Neural Computing: Theory and Practice** (Van Nostrand Reinhold, NY, 1989), pp. 56-58.

[11] Y. C. Lee, "Neural networks with memory for intelligent computations," in **Proceedings of the 13th Conference on the Numerical Simulation of Plasmas**, Santa Fe, New Mexico, September 17-20, 1989; J. A. Howell, C. W. Barnes, S. K. Brown, G. W. Flake, R. D. Jones, Y. C. Lee, S. Qian, and R. M. Wright, "Control of a negative-ion accelerator source using neural networks," to be published in **Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems**, Vancouver, B.C., Canada, October 30-November 3, 1989 and Los Alamos National Laboratory Rpt. No. LA-UR-89-3597 (1989).

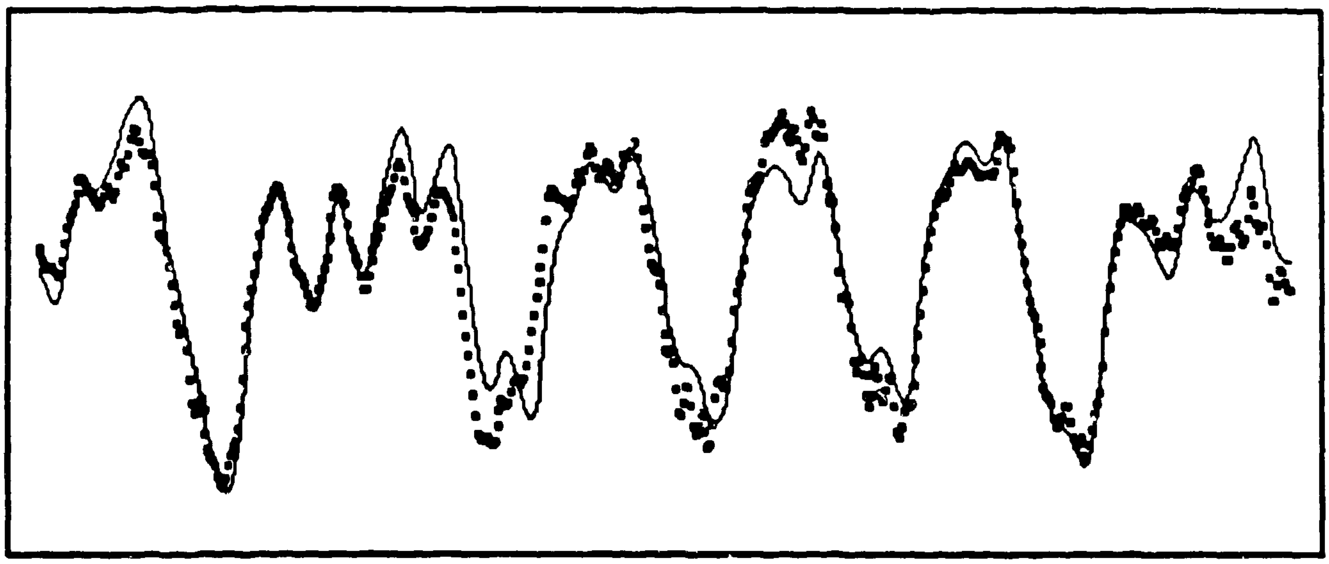

Fig. 1. The Mackey-Glass time series (solid line) and the predictions (points) of CNLS-net with 28 hidden nodes, for 500 test points. The net was trained for 40 epochs and tested at 21 iterations or $\Delta t_{pred}= 126$.

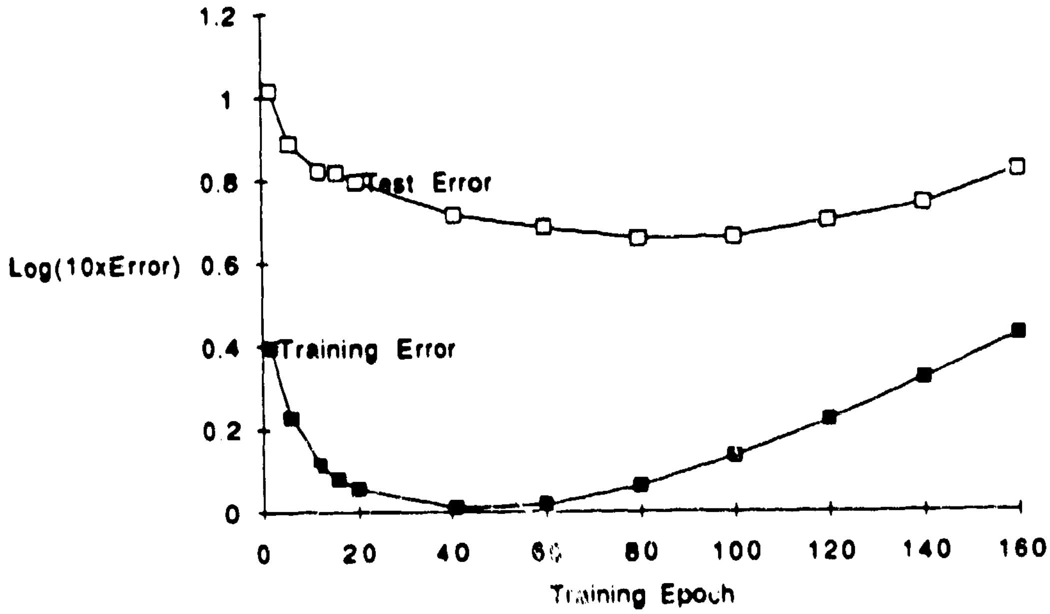

Fig. 2. Training and test error as a function of training epoch, for CNLS-net with 15 hidden nodes.

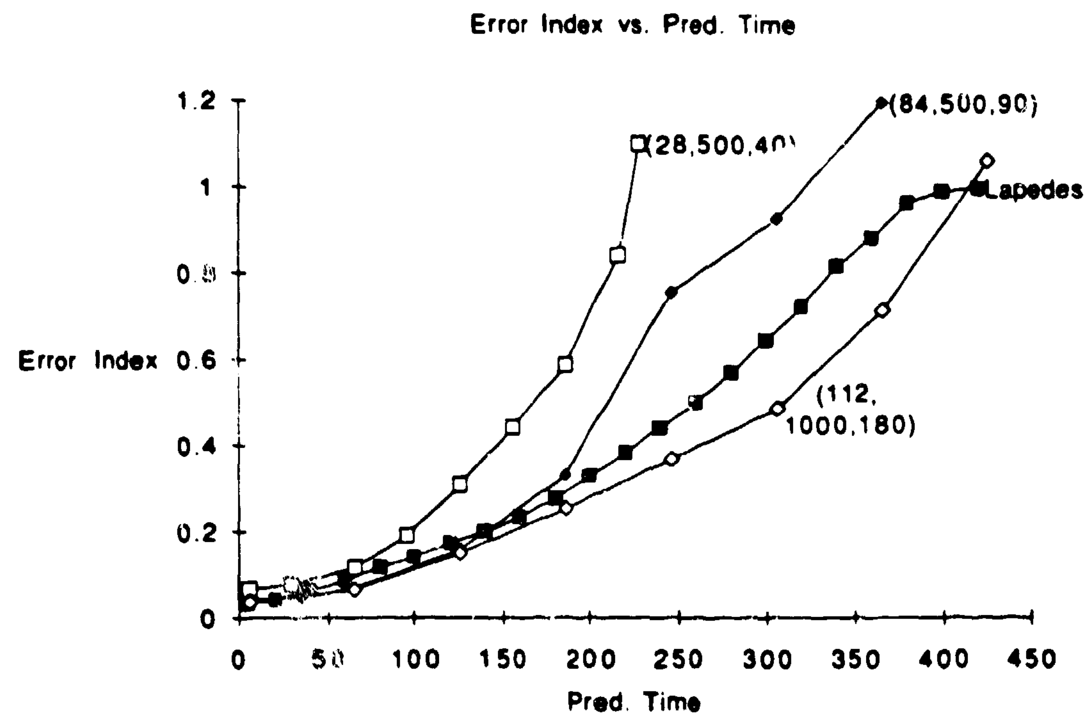

Fig. 3. Prediction error *vs.* prediction time for three CNLS-net configurations. Each CNLS-net curve is labelled by the number of hidden nodes, the number of training sets, and the number of training epochs. Curve labelled "Lapedes" is from Ref. ry6.

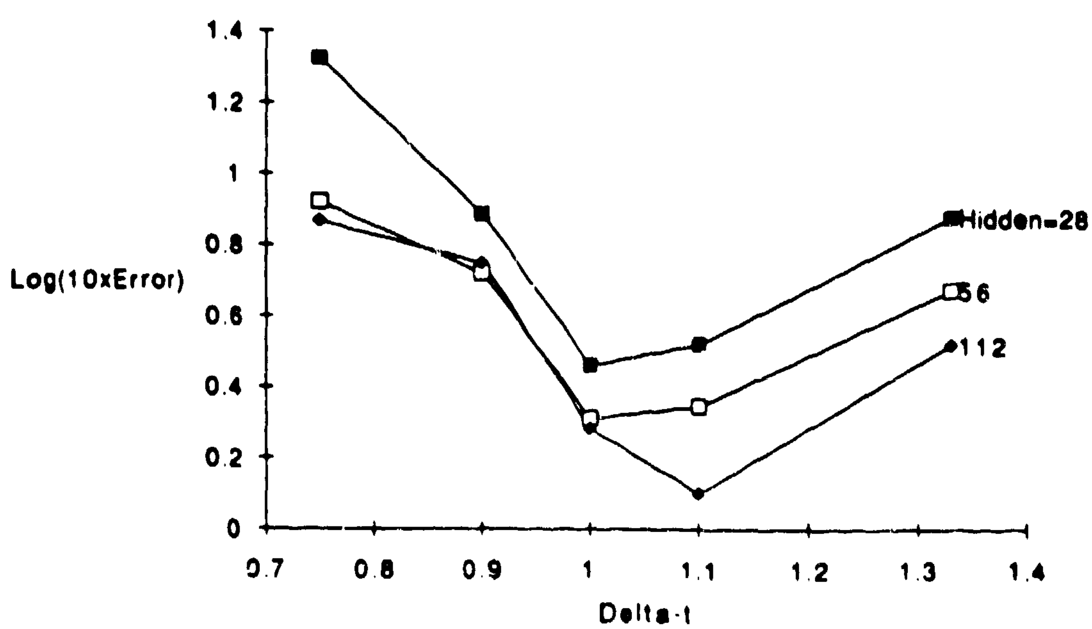

Fig. 4. Prediction error as the data files' Δt_{data} is varied shows versatility of CNLS-net. Net parameters were held fixed and net was retrained for each value of Δt_{data}.